disseminate low-value information or misinformation to users of the social network, thereby degrading their social networking experience. Applying one or more of the social media distribution policies as described above suppresses the distribution of irrelevant or low-value spam, thereby greatly enhancing the social networking experience for users.

[0052] Additionally, spammers are often very quick to adapt their spamming methods so as to bypass or circumvent spam detection filters. The "spoofing" mechanism as described above reduces the probability spammers realize their content is suppressed; even if the spammer logs in from another account and realizes that his or her media is suppressed, it is difficult for the spammer to determine what behavior is triggering the application of the social media distribution policies.

[0053] Thus, the disclosed methods may detect both images that are illegitimate as well as tags that are illegitimate to prevent distribution of spam or spam-like promotions. The two methods may be run on all images simultaneously, or at the time tag requests are received on a particular image to effectively filter a multitude of spam-like actions. In particular embodiments, the image legitimacy method is applied upon upload, and the tag analysis is performed any time tag requests are received on the image. In this way, all methods of photo tag spam may be detected and suppressed. For example, the image **410** of FIG. **4**B could possibly pass the tag spam detection method of FIG. **6**, if the tags were received an acceptable duration apart from each other, in no particular order, and the members tagged had a sufficient social relevance coefficient between them. However, image **410** would not pass the image legitimacy test of FIG. **5**. Conversely, image **420** of FIG. **4**C could pass the image legitimacy test of FIG. **5** if it were a natively-captured image from a camera. However, it would not pass the tag analysis of FIG. **5**. Thus the disclosed methods may detect and suppress legitimate images with spam tags, illegitimate images with legitimate tags, as well as illegitimate images with spam tags.

[0054] FIG. **7** illustrates an example computer system **700**. In particular embodiments, one or more computer systems **700** perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems **700** provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems **700** performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems **700**.

[0055] This disclosure contemplates any suitable number of computer systems **700**. This disclosure contemplates computer system **700** taking any suitable physical form. As example and not by way of limitation, computer system **700** may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system **700** may include one or more computer systems **700**; be unitary or distributed; span multiple locations; span multiple machines; span multiple datacenters; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems **700** may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems **700** may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems **700** may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0056] In particular embodiments, computer system **700** includes a processor **702**, memory **704**, storage **706**, an input/output (I/O) interface **708**, a communication interface **710**, and a bus **712**. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0057] In particular embodiments, processor **702** includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor **702** may retrieve (or fetch) the instructions from an internal register, an internal cache, memory **704**, or storage **706**; decode and execute them; and then write one or more results to an internal register, an internal cache, memory **704**, or storage **706**. In particular embodiments, processor **702** may include one or more internal caches for data, instructions, or addresses. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0058] In particular embodiments, memory **704** includes main memory for storing instructions for processor **702** to execute or data for processor **702** to operate on. As an example and not by way of limitation, computer system **700** may load instructions from storage **706** or another source (such as, for example, another computer system **700**) to memory **704**. Processor **702** may then load the instructions from memory **704** to an internal register or internal cache. To execute the instructions, processor **702** may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor **702** may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor **702** may then write one or more of those results to memory **704**. In particular embodiments, processor **702** executes only instructions in one or more internal registers or internal caches or in memory **704** (as opposed to storage **706** or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory **704** (as opposed to storage **706** or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor **02** to memory **704**. Bus **712** may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor **702** and memory **704** and facilitate accesses to memory **704** requested by processor **702**. Although this disclosure